2014 22nd International Conference on Pattern Recognition

# Feature Relevance for Kernel Logistic Regression and Application to Action Classification

Ouiza Ouyed
Department of Computer Science and Engineering
University of Quebec in Outaouais
Gatineau, QC, J8X 3X7, Canada.
Email: ouiza.ouyed@uqo.ca

Mohand Saïd Allili
Department of Computer Science and Engineering
University of Quebec in Outaouais
Gatineau, QC, J8X 3X7, Canada.
Email: mohandsaid.allili@uqo.ca

***Abstract*—An approach is proposed for incorporating feature relevance in mutinomial kernel logistic regression (MKLR) for classification. MKLR is a supervised classification method designed for separating classes with non-linear boundaries. However, it assumes all features are equally important, which may decrease classification performance when dealing with high-dimensional or noisy data. We propose a feature weighting algorithm for MKLR which automatically tunes features contribution according to their relevance for classification and reduces data over-fitting. The proposed algorithm produces more interpretable models and is more generalizable than MKLR, Kernel-SVM and LASSO methods. Application to simulated data and video action classification has provided very promising results compared to the aforementioned classification methods.**

***Keywords*—Multinomial kernel logistic regression, feature relevance, video action recognition.**

## I. Introduction

Kernel logistic regression (KLR) as support vector machines (SVM) [14] belong to the family of discriminative methods for classification. However, contrarily to SVM, KLR is based on probabilistic assignment of data and is more naturally extendable to multi-class classification. Feature weighting is an inherent property for linear logistic regression (LLR) [17, 11], but it is not the case in KLR where features are considered equally important for classification [17]. This can be problematic for several classification-based applications where performance is dependant on the dimensionality of data and separability between classes of observations. Examples of these applications include gene selection, text categorization, speech and video action recognition, to name a few.

Several approaches have used feature selection (FS) for improving classification performance [3]. We focus our study on sparse models using regression. These include the *least absolute shrinkage and selection operator* (LASSO) [12] and its generalized version [8], *relevance vector machines* (RVM) [13], *import vector machines* (IVM) [17] and many others [4]. The LASSO, for example, is based on linear logistic regression. It shrinks some logistic coefficients to zero, resulting in models retaining the most discriminative features for classification. In [8], an extension of LASSO has been proposed using group penalty function for logistic regression. The IVM [17] is another algorithm that searches a subset of data instances yielding the best classification. The algorithm performs better than SVM but incurs a huge computational time since it performs the search in a greedy fashion. RVM [13] performs similarly to IVM but for regression. Similarly to LASSO, a Bayesian logistic regression using $\ell_1$-norm penalty has been proposed in [2]. This method avoids data over-fitting for text classification but it is dedicated to linearly separable classes.

FS aims at building feature subsets (i.e., sparse models) that ensure optimal classification [3]. Feature weighting (FW) is another paradigm for achieving the same objective. It aims at assigning weights to features according to their performance for classification. Indeed, FW can perform even better than FS since FS can be seen as a special case of FW (e.g., by considering binary weights $\{0, 1\}$). For instance, a weighted KLR has been proposed in [9] for imbalanced and rare events classification. The approach performs better than standard KLR. However, it is geared more toward binary classification. In [15], an $\ell_0$-"norm"-based penalization has been used for feature weighting in SVM. However, the approach is limited to linear binary classification. This limitation stems from the nature of the learning machine (e.g., (SVMs [14]), where generalization to multi-class classification can not be easily formulated. Besides, a subset of features may be relevant for discriminating one class from another (e.g., for binary classification). However, when generalizing to multi-class classification, relevant features for discriminating one class may not be relevant for discriminating another class.

We propose an approach, coined fr-MKLR, incorporating feature relevance (FR) in *multinomial kernel logistic regression* (MKLR). MKLR as for SVM belong to the family of supervised methods for classification. However, it assumes all features are equally important, which may decrease classification performance when dealing with high-dimensional or noisy data. To obtain sparse classification models in MKLR, FR is directly encoded in the radial-basis function of kernels using variable weights. A penalization based on the so-called $\ell_0$-"norm" is added to the likelihood function to encourage model sparsity. Beside formulating FR for each class, obtained sparse models can deal with arbitrary numbers of classes and dimensions of data. Experiments including simulated datasets and application to human action recognition in videos have shown that fr-MKLR is computationally efficient and compares favorably to MKLR, kernel SVM and LASSO.

The rest of this paper is organized as follows: Section II describes MKLR. Section III describes the proposed fr-MKLR. Section IV provides experiments validating our approach. We end the paper with a conclusion and future work perspectives.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.237

## II. Multinomial Kernel Logistic Regression

MKLR produces non-linear classification boundaries by transforming the input variable space into another space using a positive-definite kernel $\mathcal{K}(.,.)$. In the past, the relationship between SVM and regularized function estimation in the reproducing kernel Hilbert spaces (RKHS) has been established [17]. By replacing the hinge loss function of SVM with the negative log-likelihood (NLL) of the binomial distribution, the same relation can be established with MKLR.

Let us have $n$ instances of training data $\mathbf{x}_i \in \mathbb{R}^d$, $i \in \{1, ..., n\}$, with $d$ measured features for each instance. Suppose the learning data are generated from $m$ classes ($m \geq 2$). We associate an encoding vector $\mathbf{y}_i = [y_i^{(1)}, y_i^{(2)}, ..., y_i^{(m)}]^T$ for each data point $\mathbf{x}_i$, such that $y_i^{(j)} = 1$ if $\mathbf{x}_i$ belongs to the class $j$ and $y_i^{(j)} = 0$, otherwise. Here, $[\cdot]^T$ is the vector/matrix transpose operator. For binary classification ($m = 2$), we have $y_i \in \{0, 1\}$ and fitting a decision boundary is equivalent to searching a function $f$ minimizing the NLL [17]:

$$-\sum_{i=1}^{n} y_i f(\mathbf{x}_i) + \ln\left[1 + \exp(f(\mathbf{x}_i))\right] + \frac{\lambda}{2} \parallel f \parallel^2_{\mathcal{H}_K}, \quad (1)$$

where $\mathcal{H}_K$ is the RKHS generated by $\mathcal{K}(.,.)$ and $\lambda$ controls the contribution of the regularization term smoothing $f$. Note that the NLL (1) is obtained by setting $p(y_i = 1|\mathbf{x}_i) = \exp(f(\mathbf{x}_i))/[1 + \exp(f(\mathbf{x}_i))]$ and $p(y_i = 0|\mathbf{x}_i) = 1/[1 + \exp(f(\mathbf{x}_i))]$. The optimal $f(\mathbf{x})$ has the form [5]:

$$f(\mathbf{x}) = \sum_{i=1}^{n} a_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i), \quad (2)$$

where $a_i \in \mathbb{R}$, $i = 1, ..., n$. By putting $\mathbf{a} = [a_1, ..., a_n]^T$ and $\mathbf{y} = [y_1, ..., y_n]^T$, and using the formulation (2), function (1) can be re-written in a compact form as follows [17]:

$$-\mathbf{y}^T\mathbf{K}\mathbf{a} + \mathbf{1}^T \ln\left[1 + \exp(\mathbf{K}\mathbf{a})\right] + \frac{\lambda}{2}\mathbf{a}^T\mathbf{K}\mathbf{a}, \quad (3)$$

where $\mathbf{1} = [1, 1, ..., 1]^T$ is an $n$-dimensional vector of ones and $\mathbf{K}$ is $n \times n$ matrix with $\mathbf{K}_{r,s} = \mathcal{K}(\mathbf{x}_r, \mathbf{x}_s)$, $r, s \in \{1, ..., n\}$.

When $m > 2$, we define a separate function $f_j(\mathbf{x})$ for each class $j$, where $f_j(\mathbf{x}) = \sum_{i=1}^{n} a_{ij}\mathcal{K}(\mathbf{x}, \mathbf{x}_i)$. We put the coefficients of each function $f_j$ into a vector $\mathbf{a}_j = [a_{1j}, ..., a_{nj}]^T$. Because $\sum_{j=1}^{m} p_i^{(j)} = 1$, we have $p_i^{(m)} = 1 - \sum_{j=1}^{m-1} p_i^{(j)}$. Thus, by setting $\mathbf{a}_m = \mathbf{0}$ as for LLR [18], only the parameters $\{\mathbf{a}_1, ..., \mathbf{a}_{m-1}\}$ are to be learned. For each data point $\mathbf{x}_i$, we associate a vector containing the class posterior probabilities $\mathbf{p}_i = [p_i^{(1)}, p_i^{(2)}, ..., p_i^{(m)}]^T$, where $p_i^{(j)} = p(y_i^{(j)} = 1|\mathbf{x}_i)$, $j \in \{1, ..., m\}$, $i \in \{1, ..., n\}$, defined as:

$$p_i^{(j)} = C\exp(f_j(\mathbf{x}_i)), j = 1, ..., m-1 \quad (4)$$

where $C = 1/[1 + \sum_{h=1}^{m-1}\exp(f_h(\mathbf{x}_i))]$ and $p_i^{(m)} = C$. By putting $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_{m-1}]$, the NLL for MKLR is given by:

$$\mathcal{L}(\mathbf{A}) = \sum_{j=1}^{m-1} -\mathbf{y}^{(j)^T}\mathbf{K}\mathbf{a}_j + \mathbf{1}^T \ln\left[1 + \sum_{h=1}^{m-1}\exp(\mathbf{K}\mathbf{a}_h)\right] + \frac{\lambda}{2}\sum_{j=1}^{m-1}\mathbf{a}_j^T\mathbf{K}\mathbf{a}_j, \quad (5)$$

where we define $\mathbf{y}^{(j)} = [y_1^{(j)}, y_2^{(j)}, ..., y_n^{(j)}]^T$.

## III. Feature weighting for MKLR

In what follows, we base our analysis on the Gaussian kernel defined as : $\mathcal{K}(\mathbf{x}_r, \mathbf{x}_s) = \exp\left(-\|(\mathbf{x}_r - \mathbf{x}_s)\|^2/(2\sigma^2)\right)$, with $r, s \in \{1, ..., n\}$ and $\sigma > 0$ controls the width of the kernel. Feature weighting is motivated by the fact that a good combination of features usually leads to better classification than using each feature individually. For this goal, we propose to weight features in the Gaussian kernel of the MKLR. For $m = 2$, using a weighting vector $\Psi = [\psi_1, ..., \psi_d]^T$, we have:

$$\tilde{\mathcal{K}}(\mathbf{x}_r, \mathbf{x}_s) = \exp\left(-\|(\Psi^T(\mathbf{x}_r - \mathbf{x}_s)\|^2/2\right), \quad (6)$$

which allows to express the contribution of each feature using a weighted distance. As the weight of a feature decreases, the feature's contribution to the distance computation, and therefore to classification, will be decreased.

We generalize (6) to the multi-class case by associating a feature relevance vector $\Psi_j = [\psi_{j1}, \psi_{j2}, ..., \psi_{jd}]^T$ for each class $j$, $j \in \{1, ..., m-1\}$. Thus, for each class $j$ we define a separate symmetric kernel $\tilde{\mathbf{K}}_j$ encoding the class feature relevance, with entries given as:

$$\tilde{\mathcal{K}}_j(\mathbf{x}_r, \mathbf{x}_s) = \exp\left(-\|(\Psi_j^T(\mathbf{x}_r - \mathbf{x}_s)\|^2/2\right). \quad (7)$$

To encourage model sparsity, we add a regularization on the weights $\Psi_j$ to the NLL (5) of the MKLR using the $\ell_0$-"norm" [15]. The $\ell_0$-"norm" of $\Psi_j$ is defined as $\|\Psi_j\|_0$ = card $\{k|\psi_{jk} \neq 0, k = 1, ..., d\}$, which gives the number of non-zero entries of $\Psi_j$. Note that, unlike $\ell_q$-norms with $q > 0$, $\|\cdot\|_0$ is not a norm because the triangle inequality does not hold. Since the $\ell_0$-"norm" is not smooth, it is usually approximated by the function [15]:

$$\|\Psi_j\|_0 \approx \sum_{k=1}^{d}\left[1 - \exp(-\beta\psi_{jk})\right], \quad (8)$$

where $\beta$ is an approximation parameter that can be chosen experimentally or be tuned in order to increase the performance of the classifier. Our aim is decreasing weights and contribution of noisy features to classification.

The new posterior probabilities of the classes given an observation $\mathbf{x}_i$ will be similar to those given in Eq. (4), but we substitute the kernel $\tilde{\mathbf{K}}_j$ to $\mathbf{K}$ for each class $j$. Using the $\ell_0$-"norm" penalization, the new NLL is given as follows:

$$\mathcal{L}(\mathbf{A}, \mathbf{\Psi}) = \sum_{j=1}^{m-1} -\mathbf{y}^{(j)^T}\tilde{\mathbf{K}}_j\mathbf{a}_j + \mathbf{1}^T \ln\left[1 + \sum_{h=1}^{m-1}\exp(\tilde{\mathbf{K}}_h\mathbf{a}_h)\right] + \sum_{j=1}^{m-1}\left[\frac{\lambda}{2}\mathbf{a}_j^T\tilde{\mathbf{K}}_j\mathbf{a}_j + \mu\sum_{k=1}^{d}\left[1 - \exp(-\beta\psi_{jk})\right]\right], \quad (9)$$

where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_{m-1}]$, $\mathbf{\Psi} = [\Psi_1, ..., \Psi_{m-1}]$ and $\mu$ is a regularization parameter controlling model sparsity. We use the Newton-Raphson method for minimizing the NLL (9). First, note that $\forall j \in \{1, ..., m-1\}$ and $k \in \{1, ..., d\}$, we have:

$$\partial\mathcal{L}/\partial\mathbf{a}_j = -\tilde{\mathbf{K}}_j\mathbf{y}^{(j)} + \tilde{\mathbf{K}}_j\mathbf{p}^{(j)} + \lambda\tilde{\mathbf{K}}_j\mathbf{a}_j, \quad (10)$$

$$\partial\mathcal{L}/\partial\psi_{jk} = \mathbf{c}_j^T\mathbf{Q}_{jk}\mathbf{a}_j + \mu\beta\exp(-\beta\psi_{jk}), \quad (11)$$

where we define $\mathbf{c}_j = \left(-\mathbf{y}^{(j)} + \mathbf{p}^{(j)} + \frac{\lambda}{2}\mathbf{a}_j\right)$, $\mathbf{p}^{(j)} = [p_1^{(j)}, p_2^{(j)}, ..., p_n^{(j)}]^T$ and $\mathbf{Q}_{jk} = \tilde{\mathbf{K}}_j \circ \mathbf{B}_{jk}$, with $\mathbf{B}_{jk}$ is an $n \times n$ matrix with entries defined by $\mathbf{B}_{jk}(r, s) = -\psi_{jk}(x_{r,k} - x_{s,k})^2$ and $\circ$ is the Hadamard product between matrices.

To calculate the Hessian of (9), we compute the matrices $\mathbf{M}_j$ and $\mathbf{\Phi}_j$, $j \in \{1, ..., m-1\}$, with elements defined as: $\mathbf{\Phi}_j(k,\ell) = \frac{\partial^2 \mathcal{L}(\cdot)}{\partial \psi_{jk} \partial \psi_{j,\ell}}$ and $\mathbf{M}_j(i,k) = \frac{\partial^2 \mathcal{L}(\cdot)}{\partial a_{ji} \partial \psi_{jk}}$, $k, \ell \in \{1, ..., d\}$ and $i \in \{1, ..., n\}$. Also, we define the matrix $\tilde{\mathbf{K}}^* = \text{diag}[\tilde{\mathbf{K}}_1, ..., \tilde{\mathbf{K}}_{m-1}]$, where the operator diag$[\cdot]$ builds a matrix with diagonal made of the elements of the argument. We define also the matrix:

$$\mathbf{W}^* = \begin{pmatrix} \mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \cdots & \mathbf{W}_{1,m-1} \\ \mathbf{W}_{2,1} & \mathbf{W}_{2,2} & \cdots & \mathbf{W}_{2,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{m-1,1} & \mathbf{W}_{m-1,2} & \cdots & \mathbf{W}_{m-1,m-1} \end{pmatrix} \quad (12)$$

where:

$$\mathbf{W}_{j,\ell} = \begin{cases} \text{diag}[p_1^{(j)}(1-p_1^{(j)}), ..., p_n^{(j)}(1-p_n^{(j)})] & if \quad j = \ell \\ \text{diag}[-p_1^{(j)} p_1^{(\ell)}, ..., -p_n^{(j)} p_n^{(\ell)}] & if \quad j \neq \ell \end{cases}, \quad (13)$$

We can demonstrate that the Hessian matrix of the NLL (9) can be defined as follows:

$$\tilde{\mathbf{H}} = \begin{pmatrix} \tilde{\mathbf{K}}^* \mathbf{W}^* \tilde{\mathbf{K}}^* + \lambda \tilde{\mathbf{K}}^* & \mathbf{M}^* \\ \mathbf{M}^{*T} & \mathbf{\Phi}^* \end{pmatrix}, \quad (14)$$

where $\mathbf{M}^* = \text{diag}[\mathbf{M}_1, ..., \mathbf{M}_{m-1}]$, $\mathbf{\Phi}^* = \text{diag}[\mathbf{\Phi}_1, ..., \mathbf{\Phi}_{m-1}]$. The Newton-Raphson update for estimating $\mathbf{a}_j$ and $\Psi_j$, $j \in \{1, ..., m-1\}$, is done using:

$$\begin{pmatrix} \tilde{\mathbf{a}}^{(t+1)} \\ \tilde{\Psi}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{a}}^{(t)} \\ \tilde{\Psi}^{(t)} \end{pmatrix} - \tilde{\mathbf{H}}^{-1} \tilde{\mathbf{g}}. \quad (15)$$

where $\tilde{\mathbf{\Psi}} = [\mathbf{\Psi}_1^T, \mathbf{\Psi}_2^T, ..., \mathbf{\Psi}_{m-1}^T]^T$, $\tilde{\mathbf{a}} = [\mathbf{a}_1^T, \mathbf{a}_2^T, ..., \mathbf{a}_{m-1}^T]^T$ and $\tilde{\mathbf{g}}$ is the gradient vector defined by concatenating the vectors produced by Eqs. (10) and (11).

Finally, Algorithm 1 shows the steps for estimating the parameters of our model. The algorithm ends when the estimation reaches a certain precision $\epsilon$ or a maximum number of iterations MAXITER.

**Algorithm 1** Parameter estimation for fr-MKLR method.

**Inputs**: - Data set $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$.
**Output**: - Parameter vectors $(\mathbf{a}_j, \Psi_j)$, $j \in \{1, ..., m-1\}$.

```
Ψ_j ← Ψ_j^(0); a_j ← a_j^(0);
t ← 1;
repeat
    E ← 0;
    for j = 1 → m − 1 do
        Compute ∂L/∂a_j using Eq. (10).
        Compute ∂L/∂Ψ_j using Eq. (11).
        Compute the Hessian using using Eq. (14);
        Update a_j^(t) and Ψ_j^(t) using Eq. (15);
        E ← (E + ‖a_j^(t+1) − a_j^(t)‖ + ‖Ψ_j^(t+1) − Ψ_j^(t)‖);
    end for
    t ← t + 1;
until (E < ε OR t > MAXITER)
```

## IV. Experiments

We have evaluated our method using simulated datasets and on an application to video action recognition. In each experiment, the data are divided into *learning* and *testing* parts, with number of points in each part denoted by $N_l$ and $N_t$, respectively. Obtained results using fr-MKLR are compared with MKLR, Kernel SVM, LASSO and the Naive Bayes classifier. For quantitative evaluation, classification accuracy (CA) has been used as an error measurement for comparing methods performance. The CA is given by $(1 - N_b/N_i) \times 100$, where $N_b$ is the number of points badly classified and $i \in \{l, t\}$.

### A. Tests on simulated data

To show the ability of fr-MKLR to select the best features for classification, we conducted several tests using simulated data. The training and testing data of each test have been generated using finite Gaussian mixture models (GMMs). Thus, each class data follow the following mixture model:

$$p(\mathbf{x}|y^{(j)} = 1) = \sum_{k=1}^{L_j} \pi_{j,k} p(\mathbf{x}|\mu_{j,k}, \mathbf{\Sigma}_{j,k}\}), j \in \{1, ..., m\}, \quad (16)$$

where $L_j$ is the number of components of the mixture, $\pi_{j,k}, \mu_{j,k}$ and $\mathbf{\Sigma}_{j,k}$ are the a priori probability, the mean vector and covariance matrix of the component $k$.

*1) Case of binary classification ($m = 2$):* We conducted two tests with GMMs parameters given in Table I:

- Test I: shows classification performance in case of overlapping classes. Each class is generated using one bivariate Gaussian. We vary the amount of class overlapping by shifting the mean of the second class in one dimension by increments $\tau$ of a step $\delta = 0.25$.
- Test II: shows the ability of our algorithm for generalization when learning data are scarce. Each class is generated using a mixture of bivariate Gaussians. We vary the number of generated data per Gaussian $N_g$ form 10 to 100 (see Fig. 2, right).

Figs. 1, the first and second row show two examples illustrating class boundaries obtained in Test I and Test II using MKRL and fr-MKLR, respectively. Clearly, fr-MKLR has succeeded in selecting the best separating feature which led to a better generalization than MKLR. Also, fr-MKLR performs better than LASSO and KSVM for generalization using the test data (see Fig. 2 for the obtained CA). For Test II, the average CA obtained for LASSO is $66.7\%$ and $64.5\%$ using learning and testing data, respectively. The average CA obtained for Naive Bayes is $97.73\%$ and $97.25\%$ using learning and testing data, respectively. Since we have used the true parameters of the GMMs for the naive Bayes classifier, it performed better than the other methods in the two tests.

*2) Case of multi-class classification ($m > 2$):* We conducted a multi-class experiment, Test III, with $m = 3$. Each class has been generated using a bivariate Gaussian with parameters given in Table I. The resulting classification boundaries obtained using MKRL and fr-MKLR are shown in Fig. 1 (third row). For learning data, the obtained CA for the compared methods are: $77.46\%$ for LASSO, $95.74\%$ for KSVM, $99.14\%$ for MKLR, $98.8\%$ for fr-MKLR and

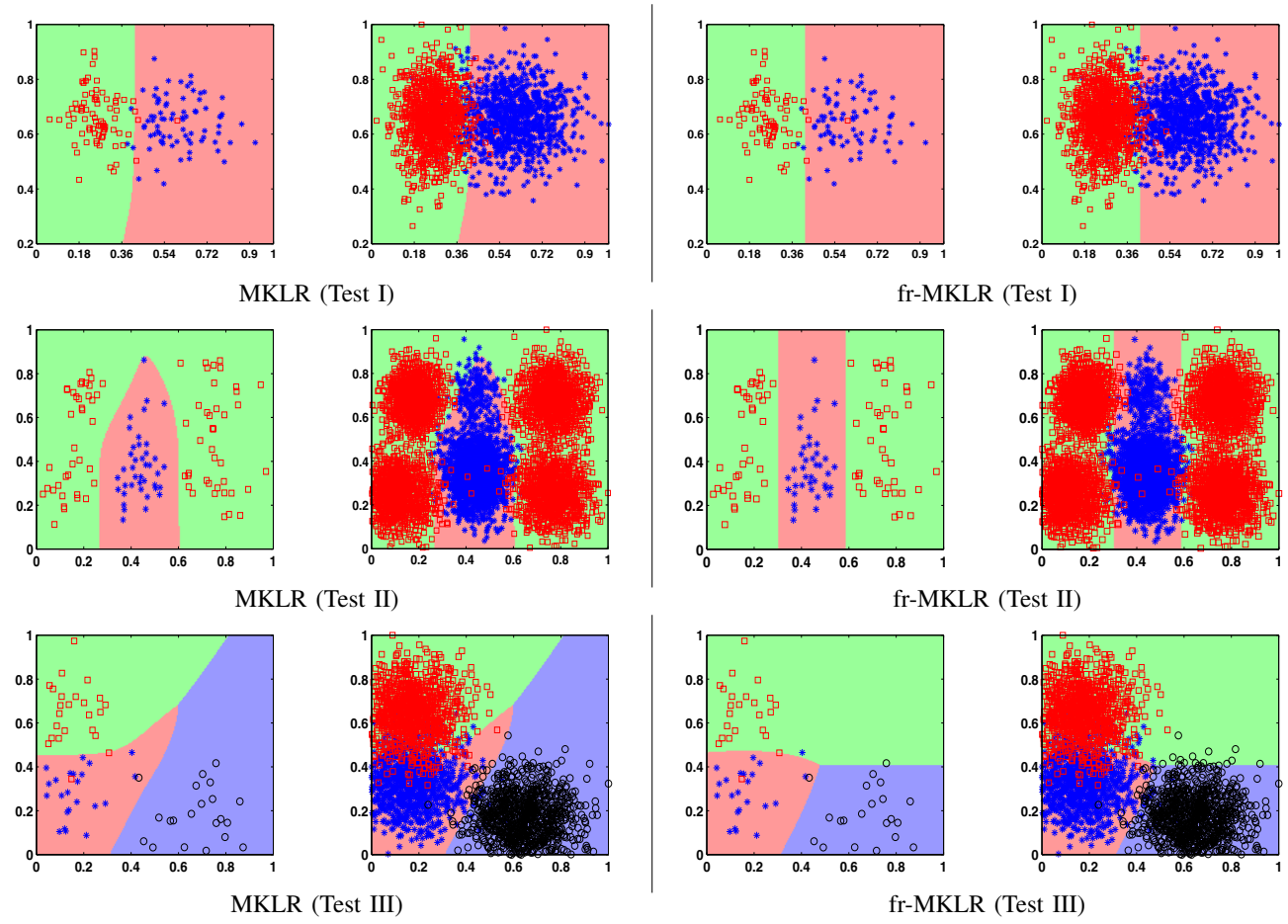


Fig. 1. Examples illustrating classification boundaries obtained by MKLR and fr-MKLR for Test I data (first row), Test II data (second row) and Test III data (third row), respectively. For each method, a 2D scatter is shown using (left) learning data and (right) testing data.

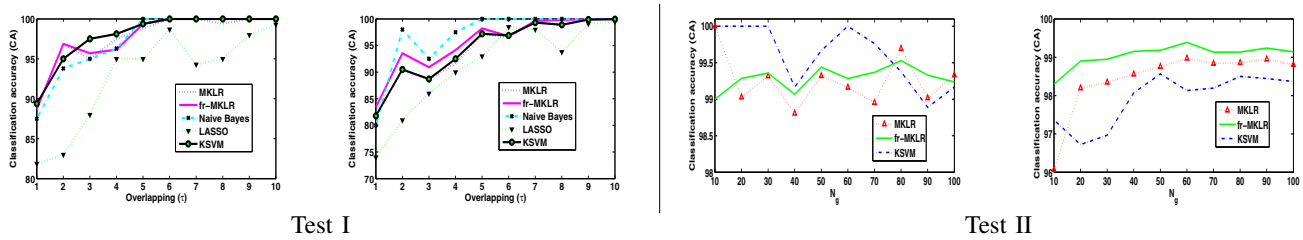


Fig. 2. Classification accuracy (CA) obtained for datasets of Test I and Test II. For each test, we show CA using (left) learning data and (right) testing data.

94.12% for Naive Bayes. For testing data, the obtained CA for the compared methods are: 75.35% for LASSO, 91.19% for KSVM, 94.89% for MKLR, 97.26% for fr-MKLR and 94.3% for Naive Bayes. Clearly, fr-MKLR has succeeded in selecting the best separating feature for each class, which led to a better generalization than the other methods.

### *B. Application for video action recognition*

The objective is classification of video actions performed by a single person. We propose a descriptor based on the shape context descriptor (SCD) [1] for representing actions spanning multiple frames of a video sequence. We start by extracting silhouettes of moving objects in each frame of the video sequence. Then, we associate a SCD of 128 entries (i.e, bins) for each silhouette contour using the silhouette center of gravity as a reference point. Each SCD entry is associated a value 0 or 1 meaning the presence (1) or absence (0) of contour in the the spatial bin corresponding to the SCD entry. The final descriptor of each action consists of the mean and standard deviation of the SCD computed using chunks of 50 frames in the sequence.

We applied our method to recognize actions using two standard datasets: KTH [6] and UIUC [7]. KTH contains actions (walking, jogging, running, boxing, hand waving, and

| Test | Class | $L_j$ | GMMs parameters |
|---|---|---|---|
| I | $j=1$ | 1 | $\mu_{1,1}=[1,2]^T$, $\mathbf{\Sigma}_{1,1}=\text{diag}([0.08,0.1])$. |
| | $j=2$ | 1 | $\mu_{2,1}=[1.75+\tau\delta,2]^T$, $\mathbf{\Sigma}_{2,1}=\text{diag}([0.2,0.08])$. |
| II | $j=1$ | 4 | $\mu_{1,1}=[1.2,4]^T$, $\mu_{1,2}=[5.2,4]^T$, $\mu_{1,3}=[5.2,1.5]^T$, $\mu_{1,4}=[0.8,1.5]^T$, $\mathbf{\Sigma}_{1,1}=\text{diag}([0.15,0.23])$, $\mathbf{\Sigma}_{1,2}=\mathbf{\Sigma}_{1,3}=\mathbf{\Sigma}_{1,4}=\text{diag}([0.28,0.28])$, $\pi_{1,1}=\pi_{1,2}=\pi_{1,3}=\pi_{1,4}=0.25$. |
| | $j=2$ | 2 | $\mu_{2,1}=[3,2]^T$, $\mu_{2,2}=[3,4.2]^T$, $\mathbf{\Sigma}_{2,1}=\text{diag}([0.18,0.33])$, $\mathbf{\Sigma}_{2,2}=\text{diag}([0.11,0.15])$, $\pi_{2,1}=0.90$, $\pi_{2,2}=0.10$. |
| III | $j=1$ | 1 | $\mu_{1,1}=[1,2]^T$, $\mathbf{\Sigma}_{1,1}=\text{diag}([0.5,0.5])$. |
| | $j=2$ | 1 | $\mu_{2,1}=[1,4]^T$, $\mathbf{\Sigma}_{2,1}=\text{diag}([0.5,0.5])$. |
| | $j=3$ | 1 | $\mu_{3,1}=[4,1]^T$, $\mathbf{\Sigma}_{3,1}=\text{diag}([0.5,0.5])$. |

TABLE I. USED GMM PARAMETERS FOR GENERATING THE DATASETS OF TEST I, II AND III, RESPECTIVELY.

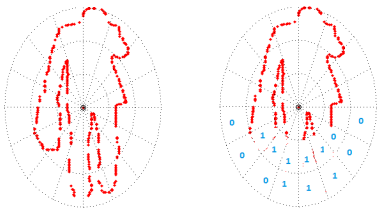

Fig. 3. Distribution of bins for the SCD used for action representation.

hand clapping) performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. UIUC consists of 14 actions (walking, running, jumping, waving, jumping jacks, clapping, jumping from sit up, raising one hand, stretching out, turning, sitting to standing, crawling, pushing up, and standing to sitting) performed by 8 actors.

We use cross validation (holdout method) for measuring the recognition performance. In each dataset, we choose 30% of data for training and 70% for testing. Table II gives CA results for these two datasets for compared methods. These include KSVM, fr-MKLR and LASSO using our action description based on SCD and the baseline methods for each dataset, namely [6] for KTH and [7] for UIUC. On average, our method fr-MKLR outperforms the others for classification. This experiment demonstrates also the performance of the proposed method for dealing with large dimensions of data.

## V. CONCLUSIONS

In this paper, we have presented an approach for introducing feature weighting in kernel logistic regression (fr-MKLR). Our method consists of assigning weights to features depending on their contribution to classification, which yields sparse models. We have applied our method to binary and multi-class data classification using simulated data and video action recognition. Experiments have shown that the proposed approach outperforms state-of-the-art methods in terms of classification accuracy.

| | Methods | Accuracy (%) | Features + classifier |
|---|---|---|---|
| KTH | laptev et al.[6] | 91.8 | Interest points+KSVM |
| | LASSO | 40 | SCD |
| | K-SVM | 76.67 | SCD |
| | fr-MKLR | **93.34** | SCD |
| UIUC | Liu et al.[7] | 93.5 | video volumes+KNN |
| | LASSO | 61 | SCD |
| | K-SVM | 92.26 | SCD |
| | fr-MKLR | **97.06** | SCD |

TABLE II. AVERAGE CLASSIFICATION ACCURACY FOR ACTION RECOGNITION OBTAINED BY THE COMPARED METHODS.

## VI. ACKNOWLEDGMENT

This work has been completed with the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

## REFERENCES

[1] S. Belongie, J. Malik and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE TPAMI*, 24(4): 509-522, 2002.

[2] A. Genkin, D.D. Lewis and D. Madigan. Large-Scale Bayesian Logistic Regression for Text Categorization. *Technometrics*, 49(3):291-304, 2007.

[3] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *JMLR*, 3:1157-1182, 2003.

[4] I. Guyon, S. Gunn, M. Nikravesh and L. Zadeh. *Feature Extraction: Foundations and Applications*. Springer, 2006.

[5] G. Kimeldorf and G. Wahba. Some Results on Tchebycheffian Spline Functions. *J. of Mathematical Analysis and Applications*, 33(1):82-95, 1971.

[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *IEEE CVPR*, 23-28, 2008.

[7] J. Liu, B. Kuipers, and S. Savarese. Recognizing Human Actions by Attributes. *IEEE CVPR*, 3337-3344, 2011.

[8] L. Meier et al., The Group Lasso for Logistic Regression. *J. of the Royal Stat. Soc.* (B), 70(1), 53-71, 2008.

[9] M. Maalouf, T.B. Trafalis. Robust Weighted Kernel Logistic Regression in Imbalanced and Rare Events Data. *Comp. Statistics & Data Analysis*, 55(1): 168-183, 2011.

[10] S. Maldonado and R. Weber. Feature Selection for Support Vector Regression Via Kernel Penalization. *IJCNN*, 1-7 , 2010.

[11] S.K. Shevade and S.S. Keerthi. A Simple and Efficient Algorithm for Gene Selection Using Sparse Logistic Regression. *Bioinformatics*, 19(17), 2246-2253, 2003.

[12] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *J. of the Royal Stat. Soc.*, B, 58(1):267-288, 1996.

[13] M.E. Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *JMLR*, 1:211-244, 2001.

[14] V.N. Vapnik. *Statistical Learning theory*. Wiley, 1998.

[15] J.Weston, A.Elisseeff and B. Schölkopf. Use of the Zero Norm with Linear Models ans Kernel Methods. *JMLR*, 3, 1439-1461. 2003

[16] N.A. Zaidi et al., Alleviating Naive Bayes Attibute Weighting Independance Assumption by Attribute Weighting. *JMLR*, 14:1947-1988, 2013.

[17] J. Zhu and T. Hastie. Kernel Logistic Regression and Import Vector Machine. *J. of Computational and Graphical Statistics*, 14(1):185-205, 2005.

[18] B. Krishnapuram et al., Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds. *IEEE TPAMI*, 27(8):957-968, 2005.